

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Matthias Müller-Hannemann
Stefan Schirra (Eds.)

Algorithm Engineering

Bridging the Gap
between Algorithm Theory and Practice

Volume Editors

Matthias Müller-Hannemann

Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik
Von-Seckendorff-Platz 1, 06120 Halle, Germany

E-mail: muellerh@informatik.uni-halle.de

Stefan Schirra

Otto-von-Guericke Universität Magdeburg, Fakultät für Informatik
Universitätsplatz 2, 39106 Magdeburg, Germany

E-mail: stschirr@ovgu.de

Library of Congress Control Number: 2010931447

CR Subject Classification (1998): F.2, D.2, G.1-2, G.4, E.1, I.3.5, I.6

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743

ISBN-10 3-642-14865-4 Springer Berlin Heidelberg New York

ISBN-13 978-3-642-14865-1 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2010

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper 06/3180

Preface

The systematic development of efficient algorithms has become a key technology for all kinds of ambitious and innovative computer applications. With major parts of algorithmic theory and algorithmic practice developing in different directions since the 1970s a group of leading researchers in the field started about 15 years ago to coin the new paradigm “Algorithm Engineering”. Its major goal is to bridge the gap between theory and practice.

This book is a collection of survey articles on different aspects of Algorithm Engineering, written by participants of a GI-Dagstuhl seminar held during September 3-8, 2006. Dorothea Wagner and Peter Sanders came up with the idea for the seminar, and approached us to organize it. In general, the concept of the GI-Dagstuhl seminars is to provide young researchers (mostly PhD students) with the opportunity to be introduced into a new emerging field of computer science. Based on a list of topics collected by the organizers, the participants prepared overview lectures they presented and discussed with other participants at the research seminar in Dagstuhl. Each contribution was elaborated afterwards and carefully cross-reviewed by all participants.

Chapter 1 gives an introduction into the emerging field of Algorithm Engineering and describes its main ingredients. It also serves as an overview for the remaining chapters of the book.

The editing process took much longer than expected, partially due to the fact that several aspects of Algorithm Engineering have never been written up before, which gave rise to lengthy internal discussions. But for the major part of the delay, the editors take their responsibility. Since the field of Algorithm Engineering has developed rapidly since the seminar took place, we made an effort to keep the contents up to date. Ideally, our book will be used as an introduction to the field. Although it has not been written as a textbook, it may well serve as accompanying material and as a reference in class.

As this book project now comes to an end, we are indebted to many people and institutions. First of all, we would like to thank the Gesellschaft für Informatik e.V. (GI) for their generous support of the GI-Dagstuhl seminar, funding the stay of all participants at Schloss Dagstuhl. We thank the Schloss Dagstuhl Leibniz-Zentrum für Informatik GmbH for their excellent workshop facilities and its hospitality, which provided the basis for a successful seminar. Alfred Hofmann and his team made it possible to smoothly publish this volume in the LNCS series of Springer. Special thanks go to Annabell Berger, Holger Blaas, and Kathleen Kletsch for their help in the editing process.

March 2010

Matthias Müller-Hannemann
Stefan Schirra

List of Contributors

Editors

Matthias Müller-Hannemann

Martin-Luther-Universität
Halle-Wittenberg
Institut für Informatik
Von-Seckendorff-Platz 1
06120 Halle, Germany
muellerh@informatik.uni-halle.de

Stefan Schirra

Otto-von-Guericke Universität
Magdeburg
Fakultät für Informatik
Universitätsplatz 2
39106 Magdeburg, Germany
stschirr@ovgu.de

Authors

Heiner Ackermann

Fraunhofer Institute for
Industrial Mathematics
Fraunhofer-Platz 1
67663 Kaiserslautern, Germany
heiner.ackermann@itwm.fraunhofer.de

Daniel Delling

Microsoft Research Silicon Valley
1065 La Avenida
Mountain View, CA 94043, USA
dadellin@microsoft.com

Deepak Ajwani

Aarhus University
MADALGO - Center for Massive
Data Algorithmics
IT-parken, Aabogade 34
8200 Aarhus N, Denmark
ajwani@cs.au.dk

Roman Dementiev

Universität Karlsruhe (TH)
Department of Computer Science,
Algorithmics II
Am Fasanengarten 5
76131 Karlsruhe, Germany
dementiev@ira.uka.de

Eric Berberich

Max-Planck-Institut für Informatik
Algorithms and Complexity
Campus E1 4
66123 Saarbrücken, Germany
eric@mpi-inf.mpg.de

Markus Geyer

Universität Tübingen
Wilhelm-Schickard-Institut für
Informatik, Paralleles Rechnen
Sand 14
72076 Tübingen, Germany
geyer@informatik.uni-tuebingen.de

Matthias Hagen

Bauhaus Universität Weimar
Faculty of Media, Web Technology
and Information Systems Group
Bauhausstr. 11
99423 Weimar, Germany
matthias.hagen@uni-weimar.de

Maria Kandyba

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl für Algorithm Engineering
Otto-Hahn-Str. 14
44227 Dortmund, Germany
maria.kandyba@cs.uni-dortmund.de

Sabine Helwig

Universität Erlangen-Nürnberg
Department of Computer Science
Hardware-Software-Co-Design
Am Weichselgarten 3
91058 Erlangen, Germany
helwig@cs.fau.de

Sascha Meinert

Karlsruhe Institute of Technology
(KIT)
Institute of Theoretical Informatics,
Algorithmics I
Am Fasanengarten 5
76131 Karlsruhe, Germany
meinert@kit.edu

Benjamin Hiller

Konrad-Zuse-Zentrum für
Informationstechnik Berlin (ZIB)
Department Optimization
Takustr. 7
14195 Berlin-Dahlem, Germany
hiller@zib.de

Henning Meyerhenke

Universität Paderborn
Department of Computer Science
Fürstenallee 11
33102 Paderborn, Germany
henningm@upb.de

Roberto Hoffmann

Martin-Luther-Universität
Halle-Wittenberg
Institut für Informatik
Von-Seckendorff-Platz 1
06120 Halle, Germany
hoffmaro@informatik.uni-halle.de

Marc Mörig

Otto-von-Guericke Universität
Magdeburg
Fakultät für Informatik
Universitätsplatz 2
39106 Magdeburg, Germany
marc@moerig.com

Falk Hüffner

Humboldt-Universität zu Berlin
Institut für Informatik
Rudower Chaussee 25
12489 Berlin, Germany
hueffner@informatik.hu-berlin.de

Hannes Moser

Friedrich-Schiller-Universität Jena
Institut für Informatik
Ernst-Abbe-Platz 2
07743 Jena, Germany
hannes.moser@uni-jena.de

Matthias Müller-Hannemann

Martin-Luther-Universität
Halle-Wittenberg
Institut für Informatik
Von-Seckendorff-Platz 1
06120 Halle, Germany
muellerh@informatik.uni-halle.de

Heiko Röglin

Maastricht University
Department of Quantitative
Economics
6200 MD Maastricht, The Netherlands
heiko@roeglin.org

Ivo Rössling

Otto-von-Guericke Universität
Magdeburg
Fakultät für Informatik
Universitätsplatz 2
39106 Magdeburg, Germany
ivo.roessling@ovgu.de

Ulf Schellbach

Technische Universität Ilmenau
Institut für Theoretische Informatik
Helmholtzplatz 1
98684 Ilmenau, Germany
ulf.schellbach@tu-ilmenau.de

Stefan Schirra

Otto-von-Guericke Universität
Magdeburg
Fakultät für Informatik
Universitätsplatz 2
39106 Magdeburg, Germany
stschirr@ovgu.de

Sven Scholz

Freie Universität Berlin
Institut für Informatik
Takustr. 9
14195 Berlin, Germany
scholz@inf.fu-berlin.de

Anna Schulze

Universität zu Köln
Zentrum für Angewandte Informatik
(ZAIK)
Gyrhofstr. 8c
50931 Köln, Germany
schulze@zpr.uni-koeln.de

Nils Schweer

Technische Universität Braunschweig
Institute of Operating Systems
and Computer Networks
Mühlenpfordtstr. 23
38106 Braunschweig, Germany
n.schweer@tu-bs.de

Johannes Singler

Universität Karlsruhe (TH)
Department of Computer Science,
Algorithmics II
Am Fasanengarten 5
76131 Karlsruhe, Germany
singler@ira.uka.de

Tobias Tscheuschner

Universität Paderborn
Department of Computer Science
Fürstenallee 11
33102 Paderborn, Germany
chessy@upb.de

Maik Weinard

Johann Wolfgang Goethe-Universität
Frankfurt am Main
Theoretische Informatik
Robert-Mayer-Str. 11-15
60325 Frankfurt am Main, Germany
weinard@thi.informatik.uni-frankfurt.de

Contents

Chapter 1. Foundations of Algorithm Engineering

M. Müller-Hannemann, S. Schirra 1

1.1	Introduction	1
1.1.1	Classical Algorithmics	1
1.1.2	The New Paradigm: Algorithm Engineering	2
1.1.3	Towards a Definition of Algorithm Engineering	4
1.1.4	Methodology	5
1.1.5	Visibility of Algorithm Engineering	6
1.2	Building Blocks of Algorithm Engineering	7
1.2.1	Modeling of Problems	8
1.2.2	Algorithm Design	9
1.2.3	Analysis	9
1.2.4	Realistic Computer Models	10
1.2.5	Implementation	11
1.2.6	Libraries	12
1.2.7	Experiments	12
1.2.8	Success Stories of Algorithm Engineering	13
1.2.9	Challenges	15
1.2.10	Further Topics — Not Covered in This Book	15

Chapter 2. Modeling

M. Geyer, B. Hiller, S. Meinert 16

2.1	Introduction	16
2.2	Modeling Fundamentals	19
2.2.1	Fundamentals	19
2.2.2	Problem Analysis	21
2.2.3	Problem Specification: Examples	23
2.2.4	Modeling a Solution Approach	26
2.2.5	Model Assessment	28
2.2.6	Inherent Difficulties within the Modeling Process	28
2.3	Modeling Frameworks	30
2.3.1	Graph-Based Models	31
2.3.2	Mixed Integer Programming	35
2.3.3	Constraint Programming	43
2.3.4	Algebraic Modeling Languages	49
2.3.5	Summary on Modeling Frameworks	53
2.4	Further Issues	53
2.4.1	Specific Input Characteristics	55
2.4.2	Problem Decomposition for Complex Applications	55
2.5	Conclusion	56

Chapter 3. Selected Design Issues	
<i>S. Helwig, F. Hüffner, I. Rössling, M. Weinard</i>	58
3.1 Introduction	58
3.2 Simplicity	60
3.2.1 Advantages for Implementation	61
3.2.2 How to Achieve Simplicity?	61
3.2.3 Effects on Analysis	65
3.3 Scalability	67
3.3.1 Towards a Definition of Scalability	68
3.3.2 Scalability in Parallel Computing	70
3.3.3 Basic Techniques for Designing Scalable Algorithms	73
3.3.4 Scalability in Grid Computing and Peer-to-Peer Networks	76
3.4 Time-Space Trade-Offs	80
3.4.1 Formal Methods	82
3.4.2 Reuse and Lookup Tables	84
3.4.3 Time-Space Trade-Offs in Storing Data	89
3.4.4 Preprocessing	92
3.4.5 Brute Force Support	93
3.5 Robustness	95
3.5.1 Software Engineering Aspects	96
3.5.2 Numerical Robustness Issues	108
3.5.3 Robustness in Computational Geometry	113
Chapter 4. Analysis of Algorithms	
<i>H. Ackermann, H. Röglin, U. Schellbach, N. Schweer</i>	127
4.1 Introduction and Motivation	127
4.2 Worst-Case and Average-Case Analysis	130
4.2.1 Worst-Case Analysis	131
4.2.2 Average-Case Analysis	132
4.3 Amortized Analysis	134
4.3.1 Aggregate Analysis	136
4.3.2 The Accounting Method	136
4.3.3 The Potential Method	136
4.3.4 Online Algorithms and Data Structures	138
4.4 Smoothed Analysis	140
4.4.1 Smoothed Analysis of Binary Optimization Problems	141
4.4.2 Smoothed Analysis of the Simplex Algorithm	151
4.4.3 Conclusions and Open Questions	158
4.5 Realistic Input Models	159
4.5.1 Computational Geometry	160
4.5.2 Definitions and Notations	162
4.5.3 Geometric Input Models	162
4.5.4 Relationships between the Models	163
4.5.5 Applications	164
4.6 Computational Testing	168
4.7 Representative Operation Counts	169

4.7.1	Identifying Representative Operations	170
4.7.2	Applications of Representative Operation Counts	171
4.8	Experimental Study of Asymptotic Performance	173
4.8.1	Performance Analysis Inspired by the Scientific Method	175
4.8.2	Empirical Curve Bounding Rules	178
4.8.3	Conclusions on the Experimental Study of Asymptotic Performance	191
4.9	Conclusions	192
Chapter 5. Realistic Computer Models		
	<i>D. Ajwani, H. Meyerhenke</i>	194
5.1	Introduction	194
5.1.1	Large Data Sets	194
5.1.2	RAM Model	196
5.1.3	Real Architecture	196
5.1.4	Disadvantages of the RAM Model	198
5.1.5	Future Trends	199
5.1.6	Realistic Computer Models	199
5.2	Exploiting the Memory Hierarchy	200
5.2.1	Memory Hierarchy Models	200
5.2.2	Fundamental Techniques	203
5.2.3	External Memory Data Structures	206
5.2.4	Cache-Aware Optimization	209
5.2.5	Cache-Oblivious Algorithms	214
5.2.6	Cache-Oblivious Data Structures	217
5.3	Parallel Computing Models	218
5.3.1	PRAM	219
5.3.2	Network Models	220
5.3.3	Bridging Models	220
5.3.4	Recent Work	223
5.3.5	Application and Comparison	225
5.4	Simulating Parallel Algorithms for I/O-Efficiency	229
5.4.1	PRAM Simulation	229
5.4.2	Coarse-Grained Parallel Simulation Results	230
5.5	Success Stories of Algorithms for Memory Hierarchies	233
5.5.1	Cache-Oblivious Sorting	233
5.5.2	External Memory BFS	233
5.5.3	External Suffix Array Construction	234
5.5.4	External A*-Search	234
5.6	Parallel Bridging Model Libraries	235
5.7	Conclusion	235

Chapter 6. Implementation Aspects

	<i>M. Mörig, S. Scholz, T. Tscheuschner, E. Berberich</i>	237
6.1	Introduction	237
6.2	Correctness	239
6.2.1	Motivation and Description	239
6.2.2	Testing	239
6.2.3	Checking	242
6.2.4	Verification	245
6.2.5	Debugging	246
6.3	Efficiency	248
6.3.1	Implementation Tricks – Tuning the Algorithms	250
6.3.2	Implementation Tricks – Tuning the Code	254
6.3.3	Code Generation	259
6.4	Flexibility	262
6.4.1	Achieving Flexibility	263
6.5	Ease of Use	267
6.5.1	Interface Design	267
6.5.2	Documentation and Readability	268
6.5.3	Literate Programming	271
6.6	Implementing Efficiently	273
6.6.1	Reuse	273
6.6.2	Programming Language	273
6.6.3	Development Environment	275
6.6.4	Avoiding Errors	275
6.6.5	Versioning	276
6.7	Geometric Algorithms	276
6.7.1	Correctness: Exact Number Types	278
6.7.2	Efficiency: Floating-Point Filters and Other Techniques	280
6.7.3	Easy to Use: The Number Types <code>CORE::Expr</code> and <code>leda::real</code>	286

Chapter 7. Libraries

	<i>R. Dementiev, J. Singler</i>	290
7.1	Introduction	290
7.2	Library Overview	292
7.3	Libraries as Building Blocks	297
7.4	Basic Design Goals and Paradigms of Combinatorial and Geometric Libraries	299
7.5	Fundamental Operations	302
7.5.1	Memory Management	302
7.5.2	Iterators versus Items	303
7.5.3	Parameterization of Data Types	304
7.5.4	Callbacks and Functors	305
7.6	Advanced Number Types	306
7.7	Basic Data Structures and Algorithms	309
7.7.1	Data Structures	309

7.7.2	Algorithms	310
7.7.3	Summary and Comparison	314
7.8	Graph Data Structures and Algorithms	314
7.8.1	Data Structures	314
7.8.2	Node and Edge Data	315
7.8.3	Algorithms	316
7.8.4	Summary and Comparison	318
7.9	Computational Geometry	319
7.9.1	Kernels and Exact Number Types	319
7.9.2	Low-Level Issues in Geometric Kernels	321
7.9.3	Functionality	322
7.9.4	Performance	323
7.10	Conclusion	324
Chapter 8. Experiments		
	<i>E. Berberich, M. Hagen, B. Hiller, H. Moser</i>	325
8.1	Introduction	325
8.1.1	Example Scenarios	325
8.1.2	The Importance of Experiments	327
8.1.3	The Experimentation Process	329
8.2	Planning Experiments	331
8.2.1	Introduction	332
8.2.2	Measures	333
8.2.3	Factors and Sampling Points	335
8.2.4	Advanced Techniques	337
8.3	Test Data Generation	339
8.3.1	Properties to Have in Mind	339
8.3.2	Three Types of Test Instances	342
8.3.3	What Instances to Use	346
8.4	Test Data Libraries	347
8.4.1	Properties of a Perfect Library	347
8.4.2	The Creation of a Library	349
8.4.3	Maintenance and Update of a Library	350
8.4.4	Examples of Existing Libraries	351
8.5	Setting-Up and Running the Experiment	353
8.5.1	Setup-Phase	354
8.5.2	Running-Phase	360
8.5.3	Supplementary Advice	364
8.6	Evaluating Your Data	367
8.6.1	Graphical Analysis	368
8.6.2	Statistical Analysis	375
8.6.3	Pitfalls for Data Analysis	381
8.7	Reporting Your Results	382
8.7.1	Principles for Reporting	382
8.7.2	Presenting Data in Diagrams and Tables	386

Chapter 9. Case Studies

<i>D. Delling, R. Hoffmann, M. Kandyba, A. Schulze</i>	389
9.1 Introduction	389
9.2 Shortest Paths	390
9.2.1 Phase I: “Theory” (1959 – 1999)	392
9.2.2 Phase II: Speed-Up Techniques for P2P (1999 – 2005)	394
9.2.3 Phase III: Road Networks (2005 – 2008)	398
9.2.4 Phase IV: New Challenges on P2P (Since 2008)	403
9.2.5 Conclusions	407
9.3 Steiner Trees	407
9.3.1 Progress with Exact Algorithms	410
9.3.2 Approximation Algorithms and Heuristics	422
9.3.3 Conclusions	426
9.4 Voronoi Diagrams	427
9.4.1 Nearest Neighbor Regions	428
9.4.2 Applications	430
9.4.3 Algorithms	431
9.4.4 The Implementation Quest	434
9.4.5 The Exact Geometric Computation Paradigm for the Computation of Voronoi diagrams	434
9.4.6 Topology-Oriented Inexact Approaches	438
9.4.7 Available Implementations	440
9.4.8 Conclusions	444

Chapter 10. Challenges in Algorithm Engineering

<i>M. Müller-Hannemann, S. Schirra</i>	446
10.1 Challenges for the Algorithm Engineering Discipline	446
10.1.1 Realistic Hardware Models	447
10.1.2 Challenges in the Application Modeling and Design Phase ...	448
10.1.3 Challenges in the Analysis Phase	449
10.1.4 Challenges in the Implementation Phase	449
10.1.5 Challenges in the Experimentation Phase	450
10.1.6 Increase the Community!	452
10.2 Epilogue	453
References	454
Subject Index	497